Introducing Bayesian Networks

- Chapter 2 from "Bayesian Intelligence"

https://bayesian-intelligence.com/publications/bai/book/BAI_Chapter2.pdf

Summary:

- Given a known set of conditional independencies, Pearl's network construction algorithm guarantees the development of a minimal network, without redundant arcs.
- BN combines qualitative information about direct dependencies (perhaps causal relations) in arcs and quantitative information about the strengths of those dependencies in conditional probability distributions.
- Computational speed gains in updating accrue when the network is sparse, allowing d-separation to take advantage of conditional independencies in the domain (so long as the Markov property holds).

Introduction to BNs

- Bayesian networks (BNs) are graphical models with nodes representing variables (discrete/continuous) and edges (arcs) representing direct connections (often causal connections) between them.
- I have RVs (nodes) $X_1, X_2, \ldots X_k$. The arcs between the RVs "--->" represent dependencies between them. If the variables are discrete, then the strength of dependencies between each node is quantified by the conditional probability distributions associated with each node.
- There must be no cycles in a BN. BN is represented by directed Acyclic Graphs (DAGs).

```
Example problem:  A man has shortness of breath (dyspnoea) and he is worry about
having lung cancer so he visits doctor.  Doctor knows that TB and bronchitis also
causes shortness of breath.  Other relevant information includes if the patient
is a smoker or the short of air pollution he is exposed to.  A positive X-ray
test would indicate either TB or cancer.
```

Three steps to build a BN

Step 1: Define nodes.

- Nodes: Identify the variables of interest of the problem.
    - What values can the variables take on?
    - Values must be mutually excusive and exhaustive-- I can only have one state at a time.
    - Boolean nodes (True or False). [Cancer]
    - Ordered nodes (low, medium, high) [Pollution]

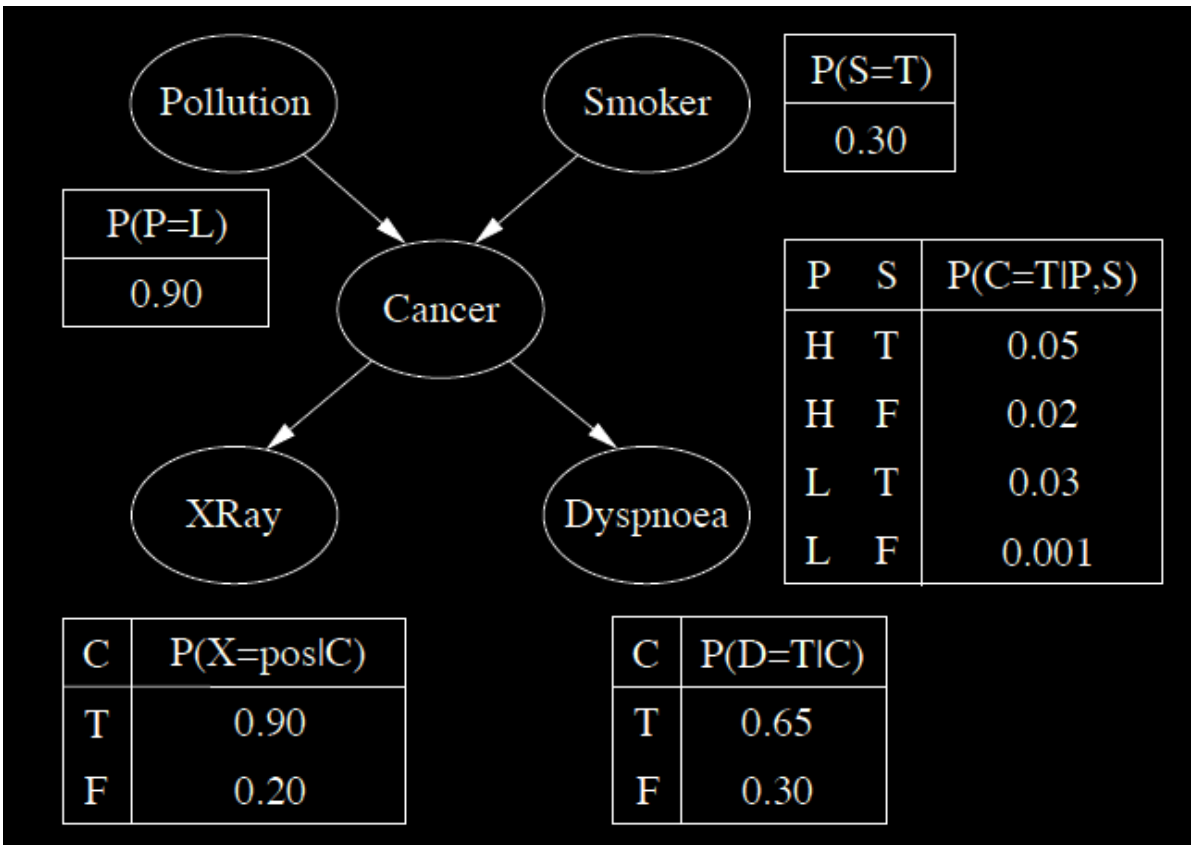
- Integral values (1…20) [Age]

Preliminary choices of nodes and values for the lung cancer example.

Node name	Type	Values
Pollution	Binary	{*low, high*}
Smoker	Boolean	{*T, F*}
Cancer	Boolean	{*T, F*}
Dyspnoea	Boolean	{*T, F*}
X-ray	Binary	{*pos, neg*}

Step 2: Define structure.

- Structure: The structure of the network captures qualitative relationships between variables.
  - Two nodes are connected directly if one causes (affects) the other.
  - The direction of effect is indicated by direction of the arrow.
  - We think that:
    - Smoking and Pollution causes cancer.
    - Having cancer will affect breathing and the chance of having a positive X-ray result.



- A node can be a parent or child:
  - Cancer has two causal **parents:** Pollution and Smoker.
  - Cancer has two **children**: XRay and Dyspnoea.
- I have a direct chain of nodes. A node is an **ancestor** of another node if it appears earlier in the chain. A node is a **descendant** of another node if it comes later in the chain.

    - Smoker is an ancestor of XRay.
    - Dyspnoea is the descendant of Pollution.
- Markov Blanket of a node: consists of the node's parents, its children, and its children's parents.
- A node with no parents (like Smoker) is a **root node.** (original cause)
- A node with no children (like XRay) is a **leaf node**. (final outcome)
- Any non-root or non-leaf nodes are **intermediate node**.

Conditional Probabilities

Step 3: Quantify relationships between connected nodes.

- Note that we are only considering discrete variables right now.
- To quantify relationships between connected nodes, we specify a conditional probability distribution for **each node**.
- With discrete variables, we do this via a conditional probability table (CPT).
- Steps:
    - For each node:
        - Look at all the possible combinations of values of its parents node. Each combination is called an "instantiation" of the parent set.
        - For each instantiation, we specify the probability that the child will take each of its values.

P	S	P(C=T|P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

- Focus on cancer node.
    - It has two parents: Pollution (P) and Smoker (S).
    - The left column contains all instantiations of the parent set.
    - In the right column, we assign the probability of cancer for each of these cases. Note for *each instantiation*, the probabilities must sum to one over *all possible states of the cancer variable.*
        - $P(C = F|P, S) = 1 - P(C = T|P, S) = 1 - 0.05 = 0.95$
- Root nodes has no parents, so its CPT only contains the prior probabilities of the node. For example $P(S = T) = 0.3$. If I randomly choose a person in the population, there will be 0.3 probability that he is a smoker. Alternatively, 30% of the population are smokers.
- For a binary node that has n parents, there are 2^{n+1} conditional probabilities!

The Markov Property

- A basic assumption of BNs.
- The Markov Property: There are no direct dependencies in the system being modelled which are not already explicitly shown via arcs in the causal graph.
    - If I know (from data) that there is conditional dependence between two variable, then there must be an arc between them in the causal graph.
    - Lack of edge implies conditional independence.
    - For example, Smoker only causes Dyspnoea through Cancer (S cannot affect D via some "backdoor").
    - BNs that satisfies the Markov Property are called Independence maps (I-maps).
- RULE: In the causal graph, no arcs = conditional independence.
- But an arc in the causal graph does not necessarily implies "real dependencies" in the system. But if every arc correspond to a direct dependence, then BN is both a I-map and a D-map (dependence map). It is called a "perfect map".

How to reason with BNs

- BN contains the full representation of the probability distributions over their variables.
- Bottom to top reasoning: A doctor observes Dyspnoea, then updates his belief about Cancer and whether the patient is a Smoker.
- Top to bottom (predictive) reasoning: A patient is a smoker, this will increase the chance of him getting cancer, and the chances of him exhibiting symptoms of cancer (such as dyspnoea).'
- Armed with a BN, and the prior/conditional probabilities of each node, I can use a number of exact and approximate inference algorithms (exists already in most BN packages) to update the belief (posterior probabilities) of each node, given some new evidence. For example, given that I *know* the patient is a Smoker (evidence), what is my posterior probability of the patient having cancer? Having a positive XRay? I can update my beliefs using causal inference algorithms.

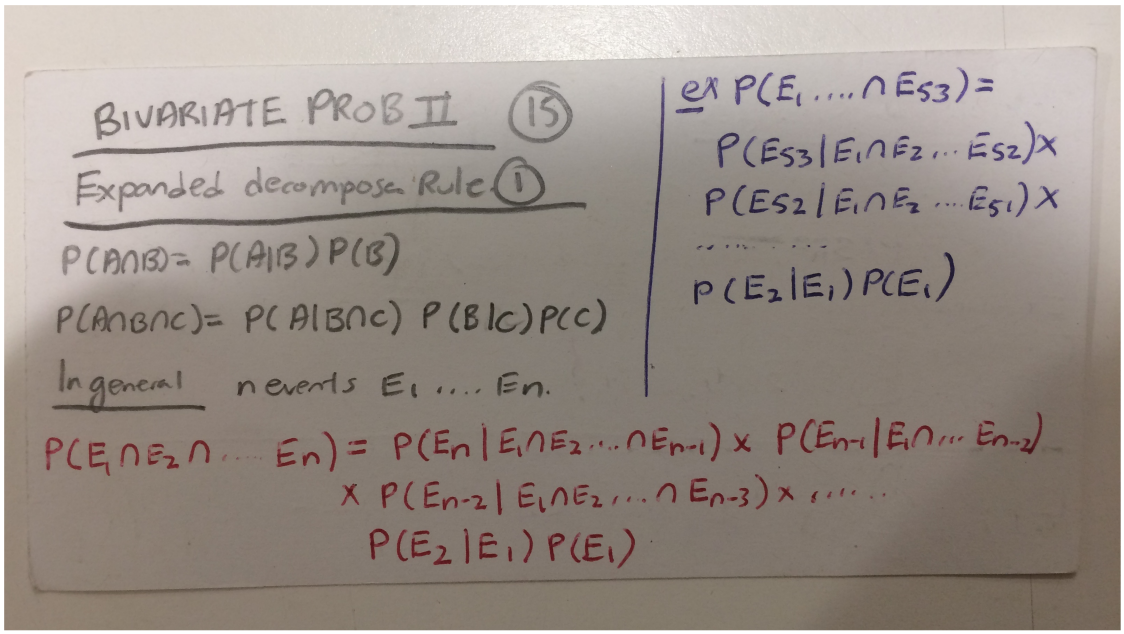
V-Structure (Intercausal reasoning)

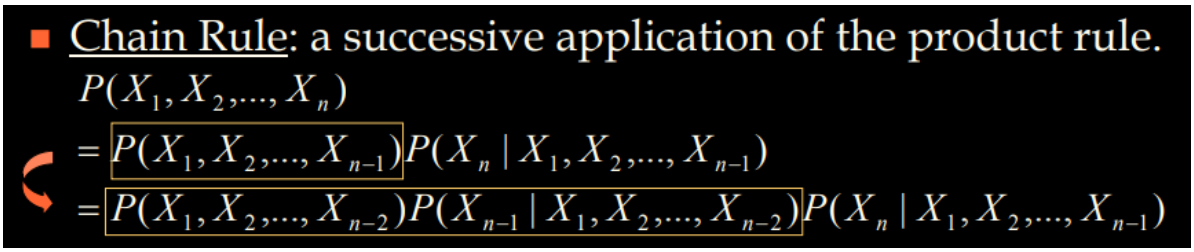
- If a node has two possible causes, then it will be represented by a V-structure in the BN.
- Cancer is caused by both Pollution and Smoker.
- Pollution and Smoker are independent-- knowing someone is living in a highly polluted area does not useful in predicting whether he is a smoker.
- However, if I know the patient has cancer, then this increase the chance that he is a smoker and exposed to pollution.
- If I know the patient has cancer, and I know he is a smoker, then the chance of him being exposed to pollution decreases, since the alternative cause is "explained away".
- **In sum, conditional on cancer, its two causes are dependent.** This is because with knowledge
  of the effect (cancer), the presence of one explanatory cause (smoker) renders an alternative cause less
  likely (pollution).

<u>Details (Semantics) about BNs</u>

- BN represents to join probability of a set of RVs in the network.

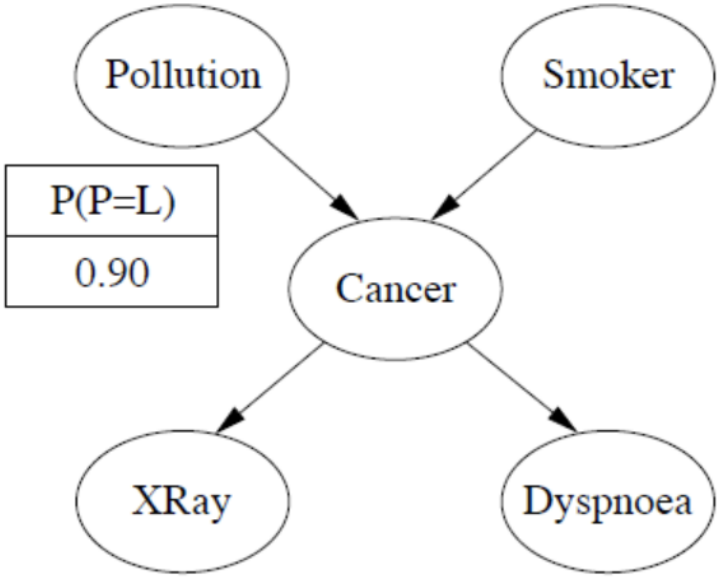
- $P(X_1 = x_1, X_2 = x_2, \ldots X_K = x_K)$
- Recall from the "Expanded decomposition rule" in probability:
-





- Thus, I can decompose
  $P(x_1, x_2, \ldots x_k) = P(x_k|x_1, \ldots, x_{k-1}) \ldots P(x_2|x_1)P(x_1) = \Pi_i P(x_i|x_1, \ldots, x_{i-1})$
- Also remember from the Markov Property of BN: the causal graph implies a node is only dependent on its parents. Thus, the value of a node is conditional only on the values of tis parents nodes. Thus, the above decomposition reduces to:
  - $P(x_1, x_2, \ldots x_k) = \Pi_i P(x_i|Parents(X_i))$
  - I only need to know a small subset of the possible conditional probabilities for each node.

Example



P(S=T)
0.30

P(P=L)
0.90

P	S	P(C=T|P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

C	P(X=pos|C)
T	0.90
F	0.20

C	P(D=T|C)
T	0.65
F	0.30

- Armed with a BN, we can find the joint probability of:

$$P(X = pos, D = T, C = T, P = low, S = F)$$

$$= P(X = pos|D = T, C = T, P = low, S = F) * P(D = T|C = T, P = low, S = F)$$

$$*P(C = T|P = low, S = F) * P(P = low|S = F) * P(S = F)$$

- To simplify the above, I only retain X|Parent(X):

$$P(X = pos, D = T, C = T, P = low, S = F)$$

$$= P(X = pos|C = T) * P(D = T|X = T) * P(C = T|P = low, S = F) * P(P = low) * P(S = F)$$

- Subbing in values from the CPT, I can compute the joint probability $P(X = pos, D = T, C = T, P = low, S = F)$.
- Thus, BN contains all information needed to compute the joint probability of the RVs in the system. BN is the representation of joint probability of the RVs in a system.
- Pearl's Construction Algorithm allow us to build a BN, given an ordered set of RVs in the system. The algorithm will add a node X to the existing network and add arcs from a minimal set of parents to the node X such that P(X|non-Parents(X)) = P(X).

Algorithm 2.1 Pearl's Network Construction Algorithm

1. *Choose the set of relevant variables* $\{X_i\}$ *that describe the domain.*
2. *Choose an ordering for the variables,* $< X_1, \ldots, X_n >$.
3. *While there are variables left:*
   (a) *Add the next variable* X_i *to the network.*
   (b) *Add arcs to the* X_i *node from some minimal set of nodes already in the net,* $Parents(X_i)$*, such that the following conditional independence property is satisfied:*

$$P(X_i|X'_1, \ldots, X'_m) = P(X_i|Parents(X_i))$$

   where $X'_1, \ldots, X'_m$ *are all the variables preceding* X_i.
   (c) *Define the CPT for* X_i.

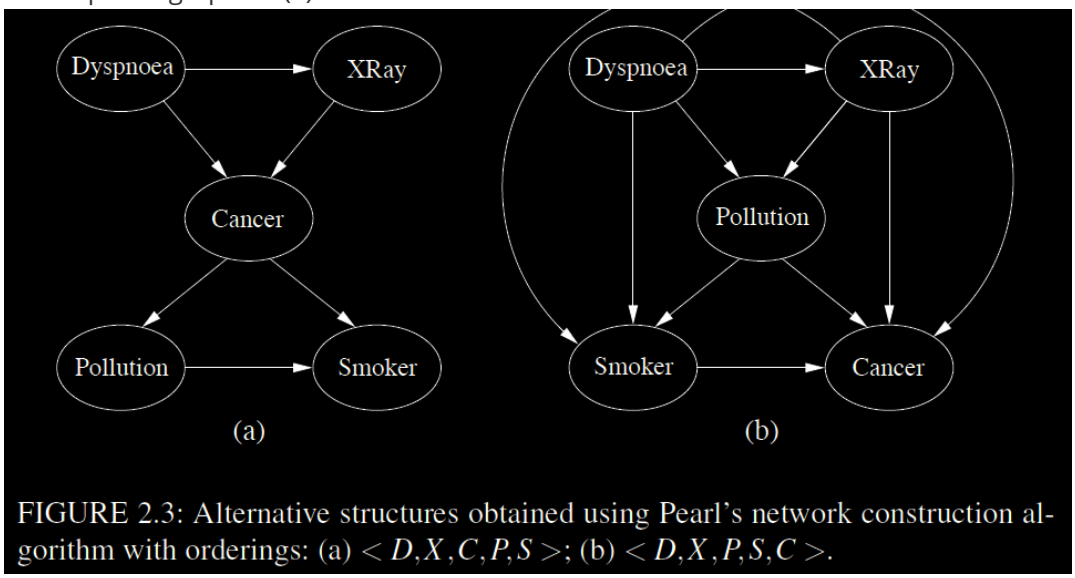
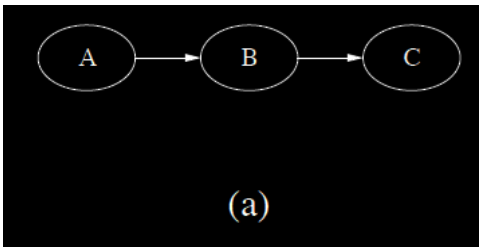
- By adding nodes in a different order, different BNs (that represents the same joint probability) can be created.
- For example, consider using <D,X,C,P,S> as input to the Peral's Network Construction Algorithm.
  - D is the root node.
  - I add X. Is X dependent (correlated) with D? Yes, since I know both D and X has a common cause (from out true causal diagram). The presence of one cancer symptom (say positive XRay) will increase the likelihood of the other (Dyspnoea). So I add arc from D to X
  - I add Cancer, I know Cancer is dependent on both Dyspnoea and XRay. So I add arc from D to C and from X to C.
  - I add Pollution. I add arc from C to P.
  - I add Smoker. I add arc from C to S. But since from the current diagram, S and P have a "common cause" in C, S and P must be dependent. So I need arc from P to S.
  - I end up with graph 2.3(a)
  - 

FIGURE 2.3: Alternative structures obtained using Pearl's network construction algorithm with orderings: (a) $< D,X,C,P,S >$; (b) $< D,X,P,S,C >$.
- We have 3 diagrams so far that represents the same joint probability. It is desirable to build the most compact BN possible.
  - I will have fewer conditional probabilities values required for model specification.
  - Probability updates will be more computationally efficient.
  - Overly dense networks do not represent the **causal dependencies of the system.**
- To get the node ordering "right" and get the most compact BN: I add the root causes first (P and S), then the variable(s) then directly influence (C), and so on until the leaves are reached.

The relationship between conditional and probabilistic (correlational) independence

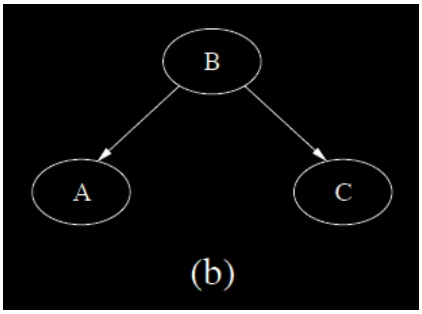
- By getting the node ordering right in Peral's network construction algorithm (starting from root causes and ending with the final outcomes), I can get a compact BN representation of the conditional (and joint) probabilities of the RV in the system. Why is that?
- If the BN satisfies Markov Property (no dependencies exist in the system that are not already represented by arcs in the DAG) =====> such BN explicitly express conditional independencies in probability distributions. SO there is a link between BN structure and conditional independence.

Causal chains

- Consider the causal chain: A cause B cause C. For example, smoking causes cancer which causes dyspnoea.
- 

- **Causal** chain => conditional independence. Specifically, $P(C|A,B) = P(C|B)$.
  - Knowing that A has occur does not make any difference to our beliefs about C, if we already know that B has occur.
  - A indep C|B
  - If I know a patient has cancer, then the probability of him having dyspnoea depends DIRECTLY only on whether they have cancer (given that he has cancer, knowing he is smoker is not useful to me in forming my belief about dyspnoea).
  - If I don't know he has canacer, then knowing that he smokes increases our belief on both cancer and dyspnoea.
  - But if I already know he has cancer, then knowing that he is a smoker c=does not change the probability of dyspnoea.
  - D is conditionally independent on S given P.

Common Causes

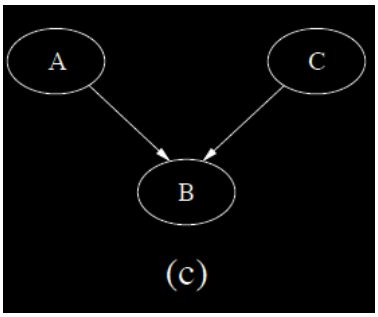
- A and C have a common cause B. For example, XRay and Dyspnoea has a common cause in Cancer.
- 

- Common causes implies the same conditional independence as causal chains:
  - $P(C|A,B) = P(C|B)$
  - A indep C |B

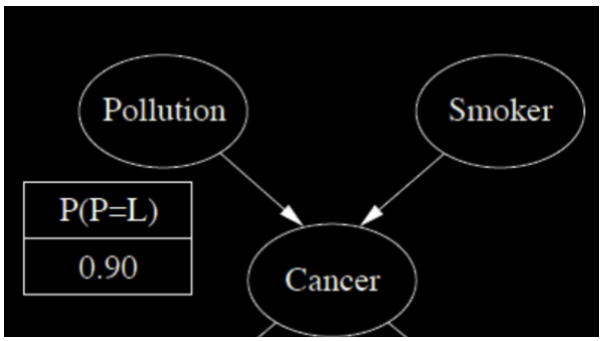
- Given I know patient has cancer, then this affects my belief on him having dyspnoea. Knowing whether he has positive XRay does not change my belief about the patient having dysponea.
- If I don't know (no evidence) that patient has cancer, then learning one symptom is present will increase the changes of cancer which in turn increase the probability of the other symptom. If I already know he has cancer, then positive XRay will not tell us anything new about the chances of dysponoea.

Common effects

- V structure. A node has two causes. For example, Cancer is caused by Pollution and Smoker.



- Common effects produces the exact opposite conditional independence to that of causal chains and common causes.
- Parents are "marginally" independent (A indep C | 0), but they become dependent conditional on the common effect (Car Start example).
- $P(A|C,B) \neq P(A|B)$
- A not indep C |B
- If I know patient has cancer, and we also know that he does not smoke, this will increase the probability of pollution.

- From the above three "rules", we see how the causal structure of the system is related to conditional probabilities amongst the RVs in the system.
- If I build a BN with a node order that **violates** the causal order of the actual system, then it will lead to additional complexity in the form of extra arcs in the BN.
- **Example**
  - 

- Consider the sub-network {Pollution, Smoker, Cancer}.
- If we build BN using the "correct" node order with root cause(s) then leaves {Pollution, Smoker, Cancer} (and Peral's algorithm), we get the simple V structure. I only have two arcs and a total of 6 probabilities (in 3 CPTs) to specify the network.
- If we build using {Cancer, Pollution, Smoker},

- We connect Pollution to Cancer since they are dependent.
- We connect Smoker with Cancer since they are dependent (because in reality there is direct dependency there). But we also need to connect Pollution with Smoker otherwise they will have an incorrect "common cause" in this graph.
- Suppose I don't add an arc between Pollution and Smoker. That means I am implying (from Markov) conditional independence between Pollution and Smoker:
  - $P(Pollution|Cancer, Smoker) = P(Pollution|Cancer)$
  - Pollution indep Smoker | Cancer
  - If that's true, then the direction of the arrows must point from Cancer to Pollution and Cancer to Smoker, which is NOT TRUE.
  - For this reason, I must add a "spurious arc" between Pollution and Smoker. Otherwise Cancer will be a common cause to Smoker and Pollution and this will induce a "spurious marginal dependency" between Smoker and Pollution. The extra arc is necessary to reestablish marginal independence between the two.

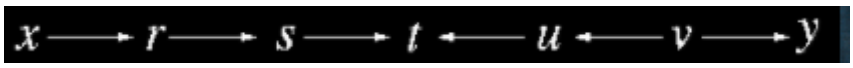
d-separation

http://bayes.cs.ucla.edu/BOOK-2K/d-sep.html#:~:text=d%2Dseparation%20is%20a%20criterion,ness%22%20or%20%22separation%22.

- d-separation is a set of rules for deciding, from a causal graph, whether a set of variables X is independent of another set Y given a third set Z. The big idea is to associate "dependence" with "connectedness" on the graph, and "independence" with "separation".
- I can look at d-separation by looking at single variables X, Y, and Z. To extend this to sets **X, Y, Z**, I just need to know that two sets are separated if and only if each element in one set is separated from every element in the other.

Unconditional separation

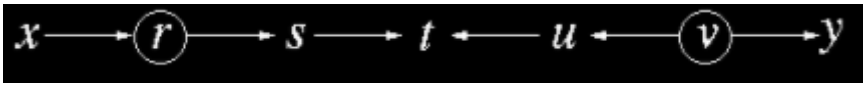
- **Rule 1:** *x* and *y* are *d*-connected if there is an unblocked path between them. By a "path" we mean any consecutive sequence of edges, disregarding their directionalities.
- By "unblocked path" we mean a path that can be traced without traversing a pair of arrows that collide "head-to-head". In other words, arrows that meet head-to-head do not constitute a connection for the purpose of passing information, such a meeting will be called a "collider".



- There is a collider at t.
- x-r-s-t is an unblocked path; thus x and t is d-connected (rule 1).
- y-v-y-t is an unblocked path; thus y and t is d-connected (rule 1).
- There is no unblocked path that I can trace from x to y; thus x and y is d-separated.
- x and y is marginally independent (cov (x,y) = 0).

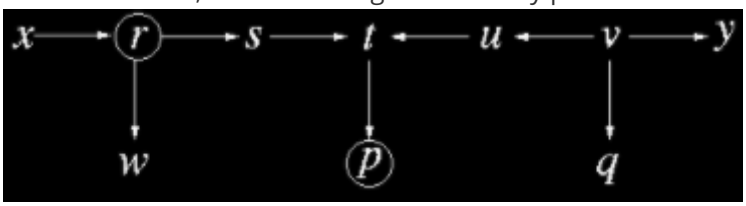
Blocking by conditioning

- We measure a set of variables Z. The conditional probability distribution of the remaining variables changes in character as the result-- some dependent pairs of variables will become independent and vice-versa.
- We look at d-connectedness, conditioned on a set of Z measured variables.

- **Rule 2:** x and y are d-connected, conditional on a set of nodes Z. This is true if there is a collider free path between x and y that traverses no member of Z. If no such path exists, we say x and y are d-separated by Z. We also say that every path between x and y is "blocked" by Z.
- 

- Z = {r,v}
- From Rule 2:
    - x and y are d-separated by Z.
    - x and s are d-separated by Z.
    - u and v are d-separated by Z.
    - u and s are d-separated by Z.
- The only paths that are d-connected conditioned on Z is s-t and u-t.
- Note that the path s-t-u is "blocked by Z" even though none of these variables belong in Z because t is a collider (and is blocked by Rule 1 and 2).

Conditioning on colliders

- V-structure: two independent variables with a common effect becomes dependent if the common effect is measured (known) due to "explained away".
- Colliders are essentially common effects.
- If we condition on colliders, or their descendants (the effects of common effects), we need a special treatment in determining d-separateness.
- **Rule 3:** If a collider (common effect) is a member of the conditioning set Z, or has a descendant in Z, then it no longer blocks any path that traces this collider.
- 

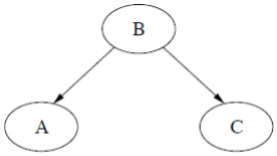
- Z = {r,p}
- s and y are d-connected by Z. Even though there is a collider in the path between s and y, the collider (t) has a descendent (p) in Z. Thus, the path s-t-u-v-y is unblocked from Rule 3 (the entire path that traverse to the collider at t is unblocked). I can flip the arrow of v-u and u-t.
- x and u are still d-separated by Z, this is because even though s-t-u is unblocked by Rule 3, the path x-r-s is blocked from Rule 2.

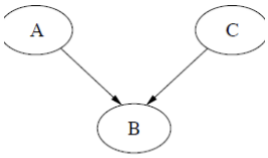
Example

$y = c1\ p + c2\ r + c3x$

- I know in the above regression model, c3 is zero. This is because x and y are d-separated given p and r (both in set Z). Thus, the corr(x,y|p,r) = 0
- y and p is d-separated by Z={r,x}? No! So c1 is not zero.
- y and r is d-separated by Z={p,x}? No! So c2 is not zero.

d-separation from Chapter 2 of Bayesian AI

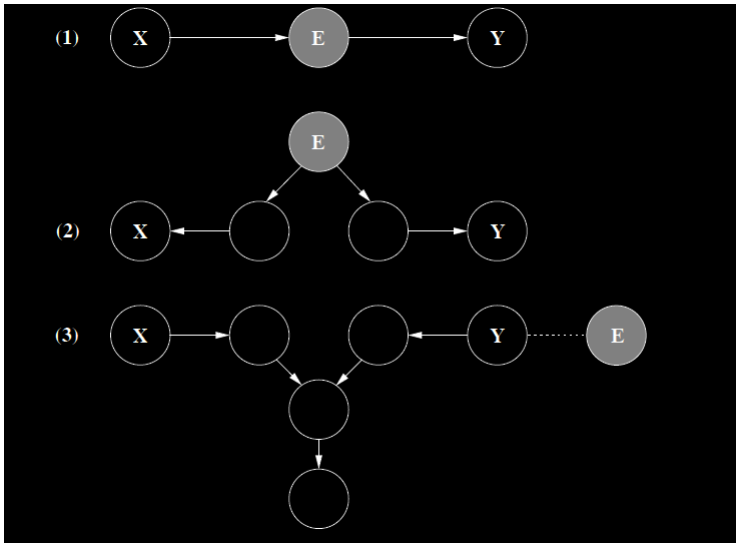
- **BN** represents conditional independencies, these independencies affect belief change during updating.
- 

(b)
- A indep C|B. Knowing B will "block" information about C being relevant to A.
- 

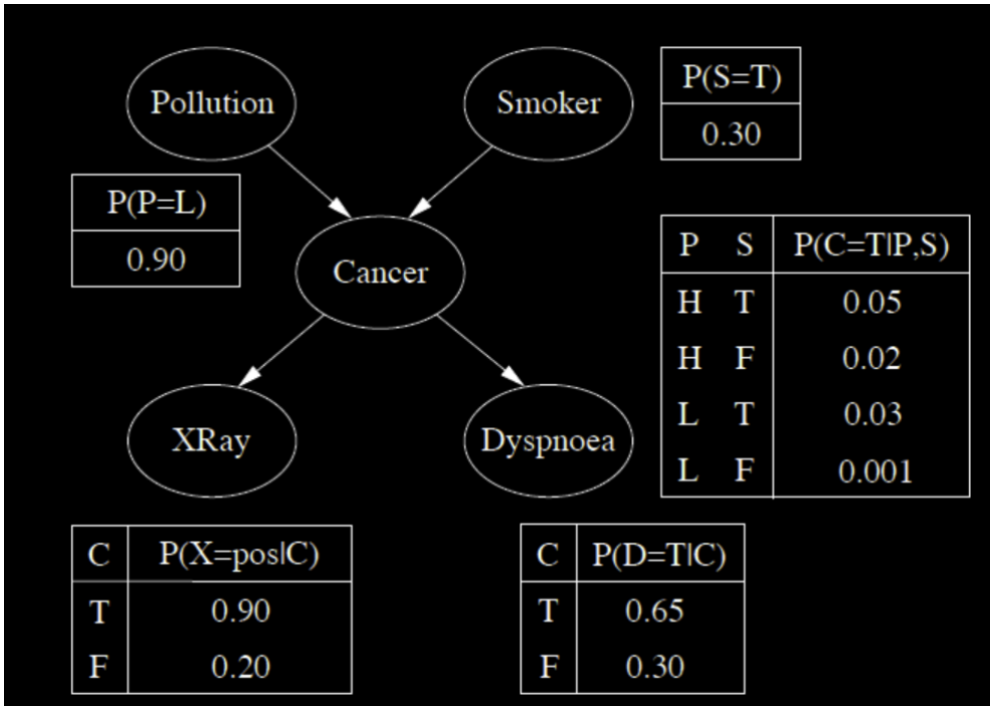
(c)
- A not indep C|B. Knowing B "activates" the relationship between A and C.

- These rules applies to sets of nodes in addition to pairs of nodes.
    - Given a set of evidence nodes **E**, it is possible to determine whether a set of nodes **X** is independent of another set **Y**. To do this, we need "d-separation".

- **Definition: PATH**
    - A path between two set of nodes **X** and **Y** is any sequence of nodes between a member of **X** and a member of Y such that every adjacent pair of nodes is connected by an arc (regardless of direction) and no node appears in the sequence twice.
- **Definition: BLOCKED PATH**
    - A path is blocked, given a set of nodes **E**, if there is a node Z on the path for which at least one of 3 conditions holds:
        - Z is in **E** and Z has one arc on the path leading in and one arc out (chain).
        - Z is in **E** and Z has both path arcs leading out (common cause).
        - Neither Z or any descendant of Z is in **E**, and both path arcs lead in to Z (common effect).
- **Definition: d-separation**
    - A set of nodes **E** d-separates two other sets of nodes **X** and **Y** (X indep Y | E) if every path from a node in X to a node in Y is blocked given E.
    - If X and Y are d-separated by E, then X and Y are conditionally independent given E (given the Markov property).
- Examples of the three BLOCKED PATH condition:
-



- Consider our example:



- P is d-separated from X and D. S is d-separated from X and D (blocking condition 1).
- X is d-separated from D (blocking condition 2).
- C, X, and D are not observed. S is d-separated from P.

Definition: d-connection

If there is a path from a node in X to a node in Y which is not blocked given E.

Look at the 3 examples from Chapter 2.